Supervisory Control via AR for Robotic Manipulation: An In-Space Assembly Use Case

Ioannis Marios Stavropoulos[1], Carol Martinez[2], Daniel J. Finnegan[1], and Juan David Hernández[1]

Abstract—**Shared autonomy in the form of shared control has been a major component in the teleoperation of Space Manipulation Systems (SMS). As demands evolve for space missions in regards to the complexity of the tasks assigned to remote robots, and bandwidth and communication delays challenges becoming harsher, it is inevitable to shift to a more autonomous supervisory control scheme in which only intermittent operator input is required. Under such constraints, it is critical to find appropriate modalities that can enable an efficient interaction with a remote robotic system. To do so, we propose to provide high-level commands to the robot via Augmented Reality (AR). We present a proof of concept and explain how we will be extending our work to a system that fully satisfies these requirements.**

I. INTRODUCTION

In-space assembly (ISA) capabilities are crucial for future space exploration and commercialization. Industry trends highlight the pivotal role of Space Manipulation Systems (SMS) in enabling scalable ISA activities such as the construction, maintenance, and upgrade of space infrastructure [1]. Robotic manipulation provides the mechanical dexterity required for precise and repetitive assembly tasks, thus reducing the need for costly and risky human extravehicular activities (EVA). As the demand to build larger and more complex structures increases, the need for autonomous SMS also increases. Although full (one-to-one) teleoperation through shared control strategies has been fundamental for SMS operations from space stations to servicing and exploration missions [2]–[4], its inherent limitations (communication delays, restrictive communication windows, bandwidth constraints, low-fidelity operator feedback and human errors) pose significant risks in ISA scenarios and motivate the need for higher levels of autonomy [5].

To overcome the aforementioned limitations of full teleoperation in space settings, we propose an alternative supervisory control approach and we present a proof of concept of an assembly task. As opposed to previous work that used shared control strategies [2]–[4], in our approach an operator can command a remote semi-autonomous robotic arm to assemble a modular solar panel structure by only providing high-level commands while immersed in an Augmented Reality (AR) environment. The simulated version of the Franka Emika robot was used to demonstrate our proof of concept and prove its feasibility. We build on our previous work [6], in which we introduced the use of AR for high-level task specification.

II. PROOF OF CONCEPT

A. System Overview

The operator wears an AR Head-Mounted Display (HMD) which enables them to interface with the robot and see a visualization of the task progress. The robot scans its environment and transmits the least amount of data necessary to virtually reconstruct it in AR in a model-based manner, rather than a live camera or point-cloud feed. The operator changes the state of the virtual environment to define and send a goal to the robot. Then, the robot proceeds to semi-autonomously engage with the real environment to reach the specified goal state. In the meantime, the operator supervises task execution, observing the movements of the virtual robot and the manipulated objects in the AR environment as they happen. In future versions, the operator will be able to intervene to correct errors if needed.

B. Architecture

The system consists of two main components: the semi-autonomous manipulator that uses the Robot Operating System (ROS), and the operator's AR interface made in Unity (see Fig. 1). The former comprises the robot and its environment, plus the functionality to process and execute the goals received from the operator through AR. The MoveIt! framework is used for motion planning and manipulation, and to interface with the robot. In particular, ROS nodes use it to control the joints of the robot, get information about the robot's current state (e.g. end effector (EEF) pose, planning status), move the EEF, grasp/release an object, and maintain a planning scene for collision-checking. Tracked objects (structure parts) are marked with AprilTags, which are temporarily used to simplify perception for the purposes of this proof of concept. Each tag is associated to an object and is assigned a single EEF grasp pose relative to it, which is used when grasping an object. AprilTags are also useful in calculating the last known poses of objects and referencing information about their assigned objects, including static (e.g. the object's name) and runtime (e.g. the pose of the EEF when the tag was last detected).

Unity is used to capture the AR modalities of the Magic Leap 2 (ML2) HMD for visualization and interaction with the Franka Emika Panda robot. Currently, the ROS server and ML2 are connected with the ROS TCP Endpoint as they are both part of the same LAN.

This work has been supported by the School of Computer Science and Informatics at Cardiff University, Cardiff, U.K.

[1] School of Computer Science and Informatics at Cardiff University, United Kingdom. `{StavropoulosI, FinneganD, HernandezVegaJ}@cardiff.ac.uk`

[2] Space Robotics Research Group (SpaceR), Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg. `carol.martinezluna@uni.lu`

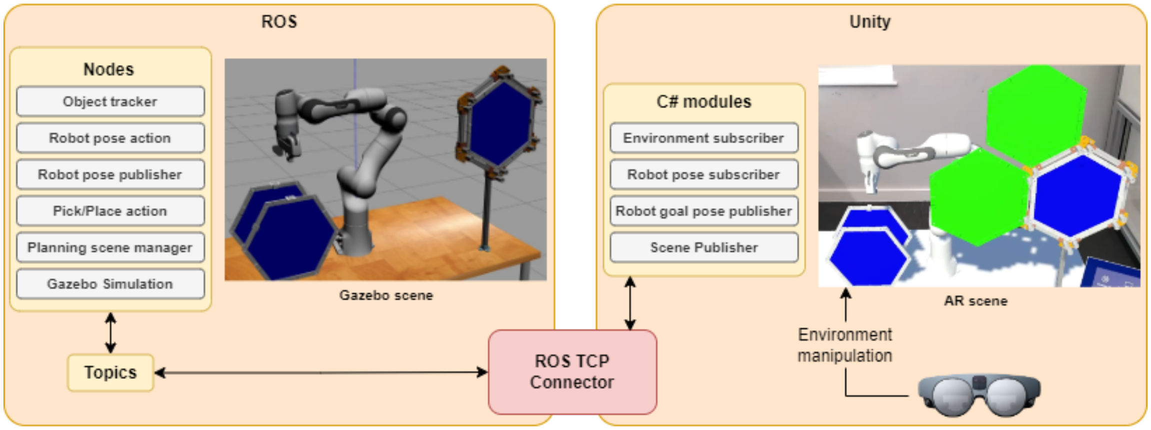

Fig. 1. Diagram of our proposed system showing its two main components. 1) ROS - A semi-autonomous manipulator robot. 2) Unity - AR environment for visualization, and publishing of high-level commands (the green placeholders are the goal poses of the two solar panel parts on the structure). ROS serialized messages are exchanged via the ROS TCP Connector.

C. *Assembly Use Case*

Once a connection between the remote (ROS) and local environments (AR) has been established, they are synchronized, and the operator is ready to give the first command to the robot, which would be to scan its environment to detect the poses of known structure parts and visualize them in AR. The robot is allowed to lose sight of an object after detecting it, as long as the environment does not change drastically. The operator reaches to grab each part from its detected position and places it appropriately on the structure, thereby specifying a goal pose. The operator then sends the goal scene to the robot, which then proceeds to semi-autonomously reconstruct it: 1) the robot moves back to the last detected pose of each of the objects' AprilTag, 2) once re-detected, its coordinate frame is used as a reference for the relative grasp pose and the current location of the object, 3) the object is grasped and placed in the specified pose while this is visualized in AR[1].

III. FUTURE WORK

Maximizing robot autonomy is a crucial component of our envisioned supervisory control scheme. We will invest heavily in a task planning aspect of the system, in an effort of delegating as much control to the robot as possible, which will allow the operator to specify goals more abstractly and minimize the need for intervention. In the occasion where error correction is required, we have plans of expanding our AR interface to accommodate such functionality in the form of intuitively visualizing errors and modifying a behavior tree online to allow task execution to proceed. Furthermore, improvements to the current methods of object tracking and perception are underway, which will consequently enhance the manipulation capabilities and dexterity of the robot. We will integrate a 'grasp regions' planner, supporting additional cameras to accommodate this feature. All of this will result in better estimates of tracked objects' poses, eliminating the need to rely on simulation data once an object has been grasped by the robot. The need for AprilTags will be further reduced as the planner will dynamically calculate multiple grasp poses for objects, rather than having a single predefined one. The additional cameras will also be used to estimate which of the multiple grasp poses is in action at any time.

IV. CONCLUSIONS

We envision the integration of a supervisory control approach in an SMS in an attempt to trivialize the challenges faced with a traditional shared control implementation, where the operator needs to provide continuous input to a remote system in real time, and where instantaneous communication is infeasible and availability is unfavorable. We have created an early adaptation of our proof of concept demonstrating our approach and have outlined the additional components we will be implementing to make the system complete.

ACKNOWLEDGMENT

Our appreciation to Akadii Kazanskii from SpaceR-SnT group who designed the 3D models of the SolarBlocks.

REFERENCES

[1] Z. Xue, J. Liu, C. Wu, and Y. Tong, "Review of in-space assembly technologies," *Chinese Journal of Aeronautics*, vol. 34, no. 11, pp. 21–47, 2021.
[2] T. Xia, S. Léonard, I. Kandaswamy, A. Blank, L. L. Whitcomb, and P. Kazanzides, "Model-based telerobotic control with virtual fixtures for satellite servicing tasks," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1479–1484.
[3] X. Li and P. Kazanzides, "Task frame estimation during model-based teleoperation for satellite servicing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2834–2839.
[4] S. Vozar, S. Léonard, P. Kazanzides, and L. L. Whitcomb, "Experimental evaluation of force control for virtual-fixture-assisted teleoperation for on-orbit manipulation of satellite thermal blanket insulation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4424–4431.
[5] P. Kazanzides, B. P. Vagvolgyi, W. Pryor, A. Deguet, S. Leonard, and L. L. Whitcomb, "Teleoperation and visualization interfaces for remote intervention in space," *Frontiers in Robotics and AI*, vol. 8, 2021.
[6] J. D. Hernández, S. Sobti, A. Sciola, M. Moll, and L. E. Kavraki, "Increasing robot autonomy via motion planning and an augmented reality interface," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1017–1023, 2020.

---

[1] Please refer to the link for a demonstration in which a simulated Franka Emika robot acts as the remote manipulator: https://youtu.be/13UA_AyZTto